(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 2 245 744 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
08.08.2012 Bulletin 2012/32

(21) Application number: **09702454.1**

(22) Date of filing: **15.01.2009**

(51) Int Cl.:
H04B 1/00 *(2006.01)* ***G06F 9/48*** *(2006.01)*
G06F 9/50 *(2006.01)*

(86) International application number:
PCT/IB2009/050137

(87) International publication number:
WO 2009/090607 (23.07.2009 Gazette 2009/30)

(54) **METHOD AND SYSTEM FOR PROCESSING RADIO PACKAGES IN A MULTIMODE SOFTWARE DEFINED RADIO (SDR) TERMINAL**

VERFAHREN UND SYSTEM ZUR VERARBEITUNG VON FUNK-PAKETEN IN EINEM MEHRMODEN-SDR-ENDGERÄT (SDR - SOFTWARE DEFINED RADIO)

PROCÉDÉ ET SYSTÈME POUR TRAITER DES PAQUETS RADIO DANS UN TERMINAL MULTIMODE DE RADIO DÉFINIE PAR LOGICIEL

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO SE SI SK TR

(30) Priority: **15.01.2008 US 21113**

(43) Date of publication of application:
03.11.2010 Bulletin 2010/44

(73) Proprietor: **NXP B.V.**
5656 AG Eindhoven (NL)

(72) Inventors:
- **HU, Liangliang**
  San Jose, CA 95131 (US)
- **SUN, Yanmeng**
  San Jose, CA 95131 (US)

(74) Representative: **Williamson, Paul Lewis**
NXP Semiconductors
Intellectual Property and Licensing
Red Central
60 High Street
Redhill
Surrey RH1 1NY (GB)

(56) References cited:
EP-A- 1 503 603
WO-A-2006/126943
US-A1- 2002 120 664
US-A1- 2003 050 055
US-A1- 2004 190 553
US-A1- 2004 242 261
US-A1- 2005 125 793
US-A1- 2006 211 387

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 2 245 744 B1

Description

[0001] The invention relates generally to wireless communications systems, and more particularly, to processing radio packages in a multimode Software Defined Radio (SDR) terminal.

[0002] SDR has been recognized as one of the most important new technologies for wireless communications in recent years. The SDR Forum (www.sdrforum.org) defines SDR technology as "radios that provide software control of a variety of modulation techniques, wide-band or narrow-band operation, communications security functions (such as hopping), and waveform requirements of current and evolving standards over a broad frequency range."

[0003] SDR technology implements signal processing functionality as software modules running on a generic processor. Furthermore, multiple software modules implementing multiple different radio protocols can be present in a single SDR terminal. Because different radio protocols require different kinds of signal processing, an SDR terminal needs to be able to schedule processing tasks related to the different radio protocols in a manner that meets the real-time requirements of each radio protocol.

[0004] Conventional task scheduling techniques include, for example, first-in first-out (FIFO), round-robin, and rate monotonic scheduling. These scheduling techniques are generic scheduling techniques that are able to handle different tasks implemented by an operating system. Implementing a flexible scheduling technique requires status monitoring and context switching, both of which take some time to perform. The status monitoring and context switching associated with typical scheduling techniques can negatively effect the time-critical processing in an SDR terminal.

[0005] In most radio protocols, the basic processing unit of a radio communication is a discrete radio package, for example, a symbol, a frame, or a time slot. With respect to an SDR terminal, baseband processing typically consists of a repetitive series of protocol-specific tasks that are performed on each radio package, for example, the number, sequence, and run time of tasks are repeated for each radio package.

[0006] The processing of a radio package usually consists of a known set of tasks and the known set of tasks is the same for each radio package. Although the set of tasks is the same for each radio package of the same radio protocol, radio packages of a different radio protocol will likely require a different set of tasks. Another characteristic of signal processing in SDR terminals is the dependency of tasks. For example, demodulating a radio package requires a set of signal processing tasks performed in a specific sequence such as channel estimation, equalization, and de-interleaving. Only after all of these tasks are finished will a radio package be demodulated successfully. If any one of the tasks in the sequence fails to be completed, the entire radio package will be lost.

[0007] In a multimode SDR terminal, different radio protocols are simultaneously supported on the same generic processing platform. In order to support simultaneous operation of different radio protocols at the same SDR terminal, a generic SDR processor must be able to interrupt a task and context switch between tasks. For example, during the processing of one radio package, if a radio package of another radio protocol arrives, the system should be able to schedule the later arriving radio package. In order to meet the time requirements of radio packages, it may be necessary to interrupt the current task processing and implement a context switch to process the later arriving radio package. Once the later arriving radio package is processed, another context switch can be made back to the original task. In order to resume processing a task, context information must be stored and reloaded back into the processing pipeline.

[0008] Most processing of incoming radio communications in SDR terminals involves pure arithmetic operations with regular memory accessing.. Soft-pipeline technology has been widely used in SDR programming to optimize the execution of program instructions. Typically, the soft-pipeline needs considerable cycles to set up and drain the pipeline upon each context switch and the sudden pipeline flush that results from an interrupt usually leads to unrecoverable data loss of the task that is in process.

[0009] Implementing an interrupt in an SDR terminal typically involves storing and uploading context information such as temporary calculation results and parameters. This requires additional resources such as registers and memory. The context store and resume operations also take some time, which can negatively affect the real time performance of a SDR terminal. Additionally, a conventional context switch causes a pipeline flush, which may cause unrecoverable data loss.

[0010] WO 2006/126943 shows an example of a programmable DSP for SDR wherein processing is divided into subtasks.

[0011] It is an object of the present invention to overcome at least some of the drawbacks of known SDRs. This object is achieved in a multimode software defined radio (SDR) terminal comprising:

a radio frequency module configured to receive a radio package;
a microprocessor configured to:

divide processing of the radio package into a series of subtasks;
process the radio package on a per-subtask basis;
drain a processing pipeline on a per-subtask basis; and

output results from the processing to a memory on a per-subtask basis, wherein the microprocessor is further configured to:

receive a radio package of a first radio protocol;
divide processing of the radio package of the first radio protocol into subtasks;
receive a radio package of a second radio protocol, wherein the first and second radio protocols are different;
divide processing of the radio package of the second radio protocol into subtasks;
perform a context switch between processing of the radio packages of the first and second radio protocols at a subtask level.

[0012] The invention is defined by the independent claims. The dependent claims describe advantageous implementations.

[0013] In another embodiment in accordance with the invention, a method for processing radio packages in a multimode SDR terminal involves receiving a radio package of a first radio protocol, dividing processing of the radio package of the first radio protocol into subtasks, receiving a radio package of a second radio protocol, wherein the first and second radio protocols are different, dividing processing of the radio package of the second radio protocol into subtasks, and performing a context switch between processing of the radio packages of the first and second radio protocols at a subtask level.

[0014] Other aspects and advantages of the present invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, illustrating by way of example the principles of the invention.

Fig. 1 illustrates a processing timeline associated with a radio package in an SDR terminal.
Fig. 2 illustrates the latest beginning time of the task from Fig. 1 given the processing time and the deadline time of the radio package from Fig. 1.
Fig. 3 illustrates a processing timeline of tasks related to two different radio packages of two different radio protocols that are processed within the same SDR terminal.
Fig. 4 illustrates a processing timeline associated with a radio package in which the processing of the radio package has been divided into subtasks.
Fig. 5 illustrates a processing timeline associated with two radio packages of two different radio protocols that are scheduled at a subtask level.
Fig. 6 is a process flow diagram of a method for processing radio packages in a multimode SDR terminal in accordance with an embodiment of the invention.
Fig. 7 is a process flow diagram of another method for processing radio packages in a multimode SDR terminal in accordance with an embodiment of the invention.
Fig. 8 depicts an embodiment of a multimode SDR terminal that includes two RF modules, a microprocessor, and memory.

[0015] Throughout the description, similar reference numbers may be used to identify similar elements.

[0016] The basic processing unit of a radio communication is a discrete radio package. Depending on the radio protocol, a discrete radio package may be identified as, for example, a symbol, a frame, or a time slot. In general, every radio package has a maximum time in which the radio package must be processed before it becomes unusable for real-time wireless communications. In a real-time system, the maximum time translates to a deadline time. If the processing of a radio package cannot be finished before the deadline time, the radio package is deemed to have failed. If the processing of the radio package is completed before the deadline time, there are processing resources that may be freed up for other uses.

[0017] Fig. 1 illustrates a processing timeline 100 associated with a radio package in an SDR terminal. In the embodiment of Fig. 1, the radio package is processed as a single task 102 and the task must be completed by a deadline time, t_D. The processing time of the task is identified as, T_P. If the processing time, T_P, is less than the deadline time, t_D, then there is some free time, T_F, and if the processing time, T_P, is greater than the deadline time, t_D, then the radio package will fail and any processing completed on the task will be lost.

[0018] Although a radio package must be completely processed by the deadline time, the latest point in time at which a task can begin and still be successfully completed is a function of the deadline time and the processing time of the radio package. Fig. 2 is a processing timeline 104 that illustrates the latest beginning time, t_L, of the task 102 from Fig. 1 given the processing time, T_P, and the deadline time, t_D, from Fig. 1. In operation, the task can begin any time before the latest beginning time, t_L, and still be completed by the deadline time. Given the above-identified illustrations, the relationship between the deadline time, t_D, the processing time, T_P, the free time, T_F, and the latest beginning time, t_L, can be expressed as:

$$t_L = t_D - T_P + T_F \qquad (1)$$

[0019] In a multimode SDR terminal, it is desirable to be able to support radio packages that are transmitted according to two different radio protocols. Further, it is desirable to be able to support the two different radio protocols simultaneously. In the case where two different radio protocols are supported simultaneously in the same multimode SDR terminal, it is possible that radio packages from the two different protocols will need to be processed during overlapping time intervals. For example, Fig. 3 illustrates a processing timeline 110 of tasks 112 and 114 related to two different radio packages of two different radio protocols that are processed within the same SDR terminal. In the situation illustrated in Fig. 3, the processing of radio package 1 has a deadline time of t_D and the processing of radio package 2 has a deadline time of t_D. Further, the processing of radio package 1 began before radio package 2 was received, where the receipt time of radio package 2 is identified as t_B, the "burst time." In order for the processing of radio package 2 to be completed before its deadline time, $t_{D'}$, the processing of radio package 2 must begin by its latest beginning time, $t_{L'}$. Beginning the processing of radio package 2 by its latest beginning time, $t_{L'}$, would require the processing of radio package 1 to be interrupted. Interrupting the processing of radio package 1 would cause all of the processing results from the current task to be lost and may cause radio package 1 to fail depending on its deadline time t_D. On the other hand, if the task for radio package 2 is not begun by its latest beginning time, $t_{L'}$, radio package 2 will fail.

[0020] In accordance with an embodiment of the invention, the processing of a radio package is divided into subtasks and results of the processing are stored on a per-subtask basis. Because the processing of a radio package is divided into subtasks and because results of the processing are stored on a per-subtask basis, context switches can be made at the subtask level instead of at the task level. With the ability to perform context switches at the subtask level, the processing of radio packages can be scheduled in a more efficient manner and context switches can be made without losing processing results that have already been generated. In addition to storing processing results on a per-subtask basis, the processing pipeline associated with the radio package is drained on a per-subtask basis. Draining the processing pipeline on a per-subtask basis can avoid the need for additional registers and memory to store data in the pipeline during context switches so that the resource utilization and complexity of context switches can be reduced.

[0021] Fig. 4 illustrates a processing timeline 120 associated with a single radio package in which the processing of the radio package has been divided into subtasks 122. In particular, processing of the single radio package is divided into subtasks ST_0 - ST_N. The processing time associated with each subtask is identified as T_{Pi} for i = 0 - N and the total processing time for an entire radio package can be expressed as:

$$T_P = \sum_{i=0}^{N} T_{Pi} \qquad (2)$$

[0022] Upon completion of each subtask, results from the subtask processing are stored in a memory so that there is no additional content in the registers of the processor. With the results of the subtask processing stored in memory, it is not necessary to backup the register content during a subsequent context switch. Additionally, a processing pipeline is drained upon the completion of each subtask. Pipelining is used in processors to allow the parallel execution of two or more consecutive instructions from a nominally sequential stream so that high efficiency processing can be achieved. However, a sudden interruption of the pipeline may induce the loss of data. In an embodiment, draining the pipeline involves finishing all of the current processing tasks that are present in the pipeline.

[0023] As described above, processing radio packages of a certain radio protocol includes a series of repetitive steps which are the same for each incoming radio package of the respective radio protocol. Therefore, the particular rules for dividing the processing of a radio package into subtasks are typically the same for each radio package of a particular radio protocol. The division of subtasks can be pre-established and applied to radio packages upon receipt at the SDR terminal.

[0024] In an embodiment, rules for dividing the processing of radio packages into subtasks are stored in a subtask database that is maintained in the SDR terminal. For example, the subtask database has at least one radio protocol-specific subtask division rule for each radio protocol that is supported by the SDR terminal. Upon receipt of a radio package, the subtask database is accessed and searched according to the radio protocol of the radio package to find the appropriate radio protocol-specific subtask division rule. In an embodiment, the subtask database is accessed the first time a radio package of a new radio protocol is received and the corresponding subtask division rule is loaded into the SDR terminal's processor. Receipt of subsequent radio packages of the same protocol does not require accessing the subtask database.

[0025] Because the processing of radio packages is divided into subtasks, context switching can be accomplished at

the subtask level instead of only at the task level. In an embodiment, context switching only occurs upon completion of a subtask. That is, a subtask is not interrupted so that a context switch can be performed. As is described below, subtask level context switching leads to better use of processing time and less lost data.

[0026] Fig. 5 illustrates a processing timeline 130 associated with two radio packages of two different radio protocols that are processed on the same SDR terminal. In the example of Fig. 5, the radio packages are the same as the radio packages of Fig. 3 except that the processing of the radio packages is divided into subtasks 132 and 134 and the scheduling of the radio packages is done at the subtask level instead of at the task level. Referring to Fig. 5, the processing of radio package 1 is divided into N subtasks, ST_0 - ST_N, and the processing of radio package 2 is divided into M subtasks, $ST_{0'}$ - $ST_{M'}$. Radio package 1 has a deadline time of t_D, radio package 2 has a deadline time of $t_{D'}$, and the processing of radio package 1 begins at time t = 0. At time t_B, radio package 2 is received at the SDR terminal in a burst. The SDR terminal then must determine whether or not it is possible to process both radio packages to completion within the deadline time of the respective radio packages, t_D and $t_{D'}$. Although it is not possible to meet the deadline time of radio package 2 if the processing of radio package 2 does not begin until the processing of radio package 1 is complete, it is possible to meet the deadline time of both radio packages 1 and 2 if the processing of radio package 1 is put on hold while radio package 2 is processed. In accordance with an embodiment of the invention, the processing of the radio packages is scheduled at the subtask level such that a first context switch takes place upon the completion of subtask ST_n of radio package 1 and a second context switch takes place upon the completion of subtask $ST_{M'}$ of radio package 2. In particular, radio package 1 is processed until the completion of subtask ST_n, at which time a context switch occurs to radio package 2. After the context switch, all of the subtasks of radio package 2 are processed to completion. Upon completion of the last subtask of radio package 2, a context switch occurs back to radio package 1. Specifically, the processing of radio package 1 resumes at subtask ST_{n+1} and the remaining subtasks of radio package 1 are then processed to completion. As illustrated in Fig. 5, subtask level context switching enables both radio protocols to be processed to completion before their respective deadline time. Additionally, because processing results are stored on a per-subtask basis, the context switching does not cause processing results to be lost.

[0027] An embodiment of subtask level scheduling in an SDR terminal is described below in conjunction with the example of Fig. 5. Referring to Fig. 5, subtask ST_n is in progress when radio package 2 is received at the SDR terminal (time t_B). Additionally, the time of completion of the current task is identified as t_S and the remaining time required to process the current radio package is T_{PR}. Given the example of Fig. 5, the time constraints of radio packages 1 and 2 are expressed as:

$$\text{Radio package 1:} \qquad t_D - t_S - T_{P'} \succ T_{PR}\,; \tag{3}$$

and

$$\text{Radio package 2:} \qquad t_{D'} - t_S \succ T_{P'}\,, \tag{4}$$

where,

$$T_{PR} = \sum_{i=n+1}^{N} T_{Pi}\ . \tag{5}$$

[0028] Before a context switch can be made from radio package 1 to radio package 2, the conditions of both equations (3) and (4) should be met. In the example of Fig. 5, at the time radio package 2 is received, the conditions of both equations are met and therefore a context switch is made from radio package 1 to radio package 2. If the conditions of both equations could not be met, the processing of radio package 1 could be continued uninterrupted.

[0029] Fig. 6 is a process flow diagram of a method for processing radio packages in a multimode SDR terminal in accordance with an embodiment of the invention. At block 200, a radio package is received. At block 202, processing of the radio package is divided into subtasks. At block 204, the radio package is processed on a per-subtask basis. At block 206, a processing pipeline is drained on a per-subtask basis. At block 208, results from the processing are stored on a per-subtask basis.

[0030] Fig. 7 is a process flow diagram of another method for processing radio packages in a multimode SDR terminal in accordance with an embodiment of the invention. At block 220, a radio package of a first radio protocol is received. At block 222, processing of the radio package of the first radio protocol is divided into subtasks. At block 224, a radio package of a second radio protocol is received, wherein the first and second radio protocols are different. At block 226, processing of the radio package of the second radio protocol is divided into subtasks. At block 228, a context switch is performed at a subtask level between processing of the radio packages of the first and second radio protocols.

[0031] The above-described techniques for processing radio packages are implemented in a multimode SDR terminal. Fig. 8 depicts an embodiment of a multimode SDR terminal 150 that includes two RF modules 152 and 154, a microprocessor 156, and memory 158. The RF modules include components for transmitting and receiving radio packages using RF signals. Components of the RF modules may include, for example, antennas, mixers, filters, oscillators, analog-to-digital converters, and digital-to-analog converters. The microprocessor is a processor that is able to execute software instructions to implement various baseband processing operations such as channel estimation, equalization, and de-interleaving. Various types of microprocessors are known in the field of SDR. The memory can include, for example, random access memory (RAM) and/or Flash memory as is known in the field. In an embodiment, the SDR terminal stores software code that when executed by the microprocessor performs the functions described above with reference to Figs. 3 and 5 - 7. Additionally, the subtask database may be stored in the memory of the SDR terminal. Although Fig. 8 depicts a multimode SDR terminal with dual RF modules, the above-described techniques are also applicable to a multimode SDR terminal that has a single RF module but still processes radio packages of different protocols in parallel.

[0032] The above-described techniques for processing radio packages in a multimode SDR terminal are applicable to SDR terminals that are compatible with various different wireless communications protocols. For example, the above-described techniques can be applied to wireless protocols that include 2G, 3GB3G cellular systems, WLAN series, WiMax, BLUETOOTH, ZigBee, Digital TV, and GPS.

[0033] Although specific embodiments of the invention have been described and illustrated, the invention is not to be limited to the specific forms or arrangements of parts as described and illustrated herein. The invention is limited only by the claims.

Claims

1. A method for processing radio packages in a multimode software defined radio SDR, terminal, the method comprising:

   receiving a radio package of a first radio protocol;
   dividing processing of the radio package of the first radio protocol into subtask (ST_0,...,ST_n;)
   receiving a radio package of a second radio protocol, wherein the first and second radio protocols are different;
   dividing processing of the radio package of the second radio protocol into subtask ($ST_{0'}$,...,$ST_{n'}$;);
   performing a context switch between processing of the radio packages of the first and second radio protocols at a subtask level.

2. The method of claim 1 wherein performing a context switch at a subtask level comprises:

   performing a subtask related to the radio package of the first radio protocol; and
   switching to a subtask related to the radio package of the second radio protocol upon completion of the subtask related to the radio protocol of the first radio package.

3. The method of claim 2 further comprising:

   performing at least one subtask related to the radio package of the second radio protocol; and
   switching to a subtask related to the radio package of the first radio protocol upon completion of a subtask related to the radio protocol of the second radio package.

4. The method of claim 2 further comprising:

   performing all subtasks related to the radio package of the second radio protocol; and
   switching to a subtask related to the radio package of the first radio protocol upon completion of all of the subtasks related to the radio protocol of the second radio package.

5. The method of claim 1 further comprising draining a processing pipeline upon performance of each subtask.

6. The method of claim 1 further comprising storing results of a subtask upon completion of each subtask.

7. The method of claim 1 wherein dividing processing of the radio packages into subtasks comprises searching a subtask database according to a radio protocol of each radio package.

8. A multimode software defined radio, SDR, terminal comprising:

a radio frequency module configured to receive a radio package;
a microprocessor configured to:

divide processing of the radio package into a series of subtasks;
process the radio package on a per-subtask basis;
drain a processing pipeline on a per-subtask basis; and
output results from the processing to a memory on a per-subtask basis, **characterized in that** the microprocessor is further configured to:

receive a radio package of a first radio protocol;
divide processing of the radio package of the first radio protocol into subtasks (ST_0,...,ST_n);
receive a radio package of a second radio protocol, wherein the first and second radio protocols are different;
divide processing of the radio package of the second radio protocol into subtasks ($ST_{0'}$,...,$ST_{n'}$)
perform a context switch between processing of the radio packages of the first and second radio protocols at a subtask level.

9. The multimode SDR terminal of claim 8 wherein the microprocessor is further configured to:

perform a subtask related to the radio package of the first radio protocol; and
switch to a subtask related to the radio package of the second radio protocol upon completion of the subtask related to the radio protocol of the first radio package.

10. The multimode SDR terminal of claim 9 wherein the microprocessor is further configured to:

process at least one subtask related to the radio package of the second radio protocol; and
switch to a subtask related to the radio package of the first radio protocol upon completion of a subtask related to the radio protocol of the second radio package.

Patentansprüche

1. Verfahren zum Verarbeiten von Funk-Paketen in einem mehrmoden-softwaredefinierten- Funk, SDR, -Terminal, das Verfahren aufweisend:

Empfangen eines Funk-Paketes eines ersten Funk-Protokolls;
Unterteilen des Verarbeitens des Funk-Pakets des ersten Funk-Protokolls in Teilaufgaben (ST_0, ..., ST_n);
Empfangen des Funk-Paketes eines zweiten Funk-Protokolls, wobei das erste und zweite Funk-Protokoll unterschiedlich sind;
Unterteilen des Verarbeitens des Funk-Paketes des zweiten Funk-Protokolls in Teilaufgaben ($ST_{0'}$,..., $ST_{n'}$);
Ausführen eines Kontext-Wechsels zwischen dem Verarbeiten des Funk-Paketes des ersten und zweiten Funk-Protokolls auf einem Teilaufgaben-Level.

2. Verfahren gemäß Anspruch 1, wobei Verarbeiten eines Kontext-Wechsels auf einem Teilaufgaben-Level aufweist:

Ausführen einer Teilaufgabe, welche sich auf das Funk-Paket des ersten Funk-Protokolls bezieht; und
Wechseln zu einer Teilaufgabe, welche sich auf das Funk-Paket des zweiten Funk-Protokolls bezieht, nach dem Fertigstellen der Teilaufgabe, welche sich auf das Funk-Protokoll des ersten Funk-Paketes bezieht.

3. Verfahren gemäß Anspruch 2 ferner aufweisend:

Ausführen zumindest einer Teilaufgabe, welche sich auf das Funk-Paket des zweiten Funk-Protokoll bezieht;

und
Wechseln zu einer Teilaufgabe, welche sich auf das Funk-Paket des ersten Funk-Protokolls bezieht, nach dem Fertigstellen einer Teilaufgabe, welche sich auf das Funk-Protokoll des zweiten Funk-Paketes bezieht.

4. Verfahren gemäß Anspruch 2 ferner aufweisend:

Ausführen aller Teilaufgaben, die sich auf das Funk-Paket des zweiten Funk-Protokolls beziehen; und
Wechseln zu einer Teilaufgabe, die sich auf das Funk-Paket des ersten Funk-Protokolls bezieht, nach dem Fertigstellen aller Teilaufgaben, die sich auf das Funk-Protokoll des zweiten Funk-Paketes beziehen.

5. Verfahren gemäß Anspruch 1 ferner aufweisend ein Entleeren einer Verarbeitungs-Pipeline nach dem Ausführen jeder Teilaufgabe.

6. Verfahren gemäß Anspruch 1, ferner aufweisend Speichern von Ereignissen einer Teilaufgabe nach dem Fertigstellen jeder Teilaufgabe.

7. Verfahren gemäß Anspruch 1, wobei das Aufteilen des Verarbeitens der Funk-Pakete in Teilaufgaben ein Durchsuchen einer Teilaufgaben-Datenbank gemäß eines Funk-Protokolls von jedem Funk-Paket aufweist.

8. Mehrmoden-software-definiertes-Funk, SDR,-Terminal, aufweisend:

ein Funk-Frequenz-Modul konfiguriert, ein Funk-Paket zu empfangen;
ein Mikroprozessor konfiguriert zum:

Unterteilen des Verarbeitens des Funk-Paketes in eine Serie von Teilaufgaben;
Verarbeiten des Funk-Paketes auf einer pro-Teilaufgaben Basis;
Entleeren einer Verarbeitungs-Pipeline auf einer pro-Teilaufgaben Basis; und
Ausgeben von Resultaten des Verarbeitens an einen Speicher auf einer pro-Teilaufgaben Basis,

dadurch gekennzeichnet, dass
der Mikroprozessor ferner konfiguriert ist zum:

Empfangen eines Funk-Paketes eines ersten Funk-Protokolls;
Unterteilen des Verarbeitens des Funk-Paketes des ersten Funk-Protokolls in Teilaufgaben (ST_0, ... ST_n);
Empfangen eines Funk-Paketes eines zweiten Funk-Protokolls, wobei das erste und zweite Funk-Protokoll unterschiedlich sind;
Unterteilen des Verarbeitens des Funk-Paketes des zweiten Funk-Protokolls in Teilaufgaben ($ST_{0'}$, ...$ST_{n'}$);
Ausführen eines Kontext-Wechsels zwischen dem Verarbeiten des Funk-Paketes des ersten und zweiten Funk-Protokolls auf einem Teilaufgaben-Level.

9. Mehrmoden-SDR-Terminal gemäß Anspruch 8, wobei der Mikroprozessor ferner konfiguriert ist zum:

Ausführen einer Teilaufgabe, welche sich auf das Funk-Paket des ersten Funk-Protokolls bezieht; und
Wechseln zu einer Teilaufgabe, welche sich auf das Funk-Paket des zweiten Funk-Protokolls bezieht, nach dem Fertigstellen der Teilaufgabe, welche sich auf das Funk-Protokoll des ersten Funk-Paketes bezieht.

10. Mehrmoden-SDR-Terminal gemäß Anspruch 9, wobei der Mikroprozessor ferner konfiguriert ist zum:

Verarbeiten zumindest einer Teilaufgabe, welche sich auf das Funk-Paket des zweiten Funk-Protokolls bezieht; und
Wechseln zu einer Teilaufgabe, welche sich auf das Funk-Paket des ersten Funk-Protokolls bezieht nach dem Fertigstellen einer Teilaufgabe, welche sich auf das Funk-Protokoll des zweiten Funk-Paketes bezieht.

Revendications

1. Procédé de traitement de paquetages radio dans un terminal radio logicielle multi-mode, SDR, le procédé comprenant:

la réception d'un paquetage radio d'un premier protocole radio ;
la division du traitement du paquetage radio du premier protocole radio en sous-tâches (ST0,..., STn) ;
la réception d'un paquetage radio d'un deuxième protocole radio, dans lequel le premier et le deuxième protocoles radio sont différents ;
la division du traitement du paquetage radio du deuxième protocole radio en sous-tâches (ST0',..., STn') ;
la réalisation d'une commutation de contexte entre le traitement des paquetages radio du premier et du deuxième protocole radio au niveau de la sous-tâche.

2. Procédé selon la revendication 1, dans lequel la réalisation d'une commutation de contexte au niveau de la sous-tâche comporte :

l'exécution d'une sous-tâche relative au paquetage radio du premier protocole radio ; et
la commutation vers une sous-tâche relative au paquetage radio du deuxième protocole radio lors de la fin de la sous-tâche relative au protocole radio du premier paquetage radio.

3. Procédé selon la revendication 2, comprenant en outre :

l'exécution d'au moins une sous-tâche relative au paquetage radio du deuxième protocole radio ; et
la commutation vers une sous-tâche relative au paquetage radio du premier protocole radio lors de la fin de la sous-tâche relative au protocole radio du deuxième paquetage radio.

4. Procédé selon la revendication 2, comprenant en outre :

l'exécution de toutes les sous-tâches relatives au paquetage radio du deuxième protocole radio ; et
la commutation vers une sous-tâche relative au paquetage radio du premier protocole radio lors de la fin de toutes les sous-tâches relatives au protocole radio du deuxième paquetage radio.

5. Procédé selon la revendication 1, comprenant en outre le vidage d'un pipeline de traitement lors de l'exécution de chaque sous-tâche.

6. Procédé selon la revendication 1, comprenant en outre le stockage des résultats d'une sous-tâche lors de la fin de chaque sous-tâche.

7. Procédé selon la revendication 1, dans lequel la division du traitement des paquetages radio en sous-tâches comporte la recherche d'une base de données de sous-tâches, en fonction d'un protocole radio de chaque paquetage radio.

8. Terminal radio logicielle multi-mode, SDR, comprenant :

un module radiofréquence configuré pour recevoir un paquetage radio ;
un microprocesseur configuré pour :

diviser le traitement du paquetage radio en une série de sous-tâches ;
traiter le paquetage radio sur la base des sous-tâches ;
vider un pipeline d'exécution sur la base de sous-tâches ; et
fournir en sortie les résultats du traitement vers une mémoire sur la base des sous-tâches, **caractérisé par le fait que** le microprocesseur est en outre configuré de manière à :

recevoir un paquetage radio d'un premier protocole radio ;
diviser le traitement du paquetage radio du premier protocole radio en sous-tâches (ST0,..., STn) ;
recevoir un paquetage radio d'un deuxième protocole radio, dans lequel le premier et le deuxième protocoles radio sont différents ;
diviser le traitement du paquetage radio du deuxième protocole radio en sous-tâches (ST0',..., STn') ;
réaliser une commutation de contexte entre le traitement des paquetages radio du premier et du deuxième protocole radio au niveau de la sous-tâche.

9. Terminal SDR multi-mode selon la revendication 8, dans lequel le microprocesseur est en outre configuré de manière à :

exécuter une sous-tâche relative au paquetage radio du premier protocole radio ; et
à commuter vers une sous-tâche relative au paquetage radio du deuxième protocole radio lors de la fin de la sous-tâche relative au protocole radio du premier paquetage radio.

10. Terminal SDR multi-mode selon la revendication 9, dans lequel le microprocesseur est en outre configuré de manière à :

exécuter au moins une sous-tâche relative au paquetage radio du deuxième protocole radio ; et
à commuter vers une sous-tâche relative au paquetage radio du premier protocole radio lors de la fin d'une sous-tâche relative au protocole radio du deuxième paquetage radio.

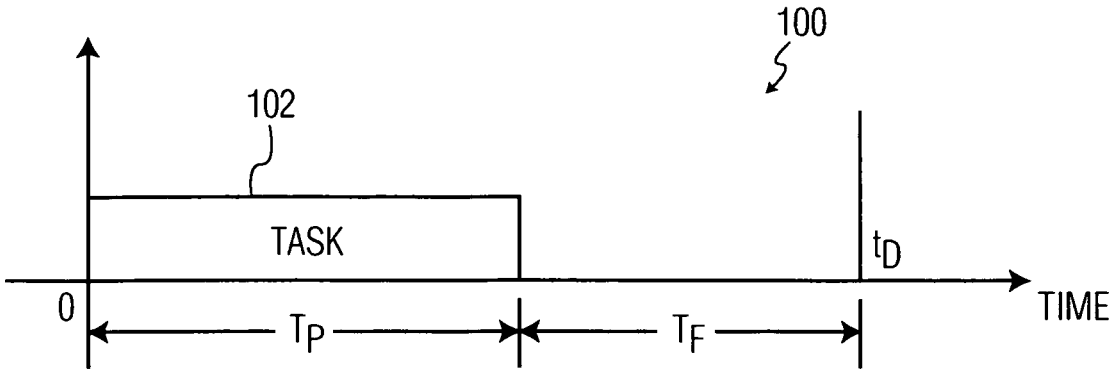

FIG. 1
PRIOR ART

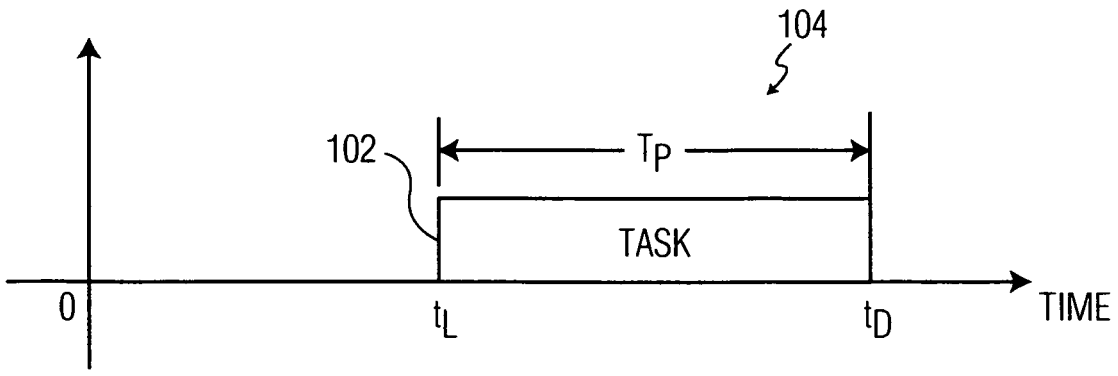

FIG. 2
PRIOR ART

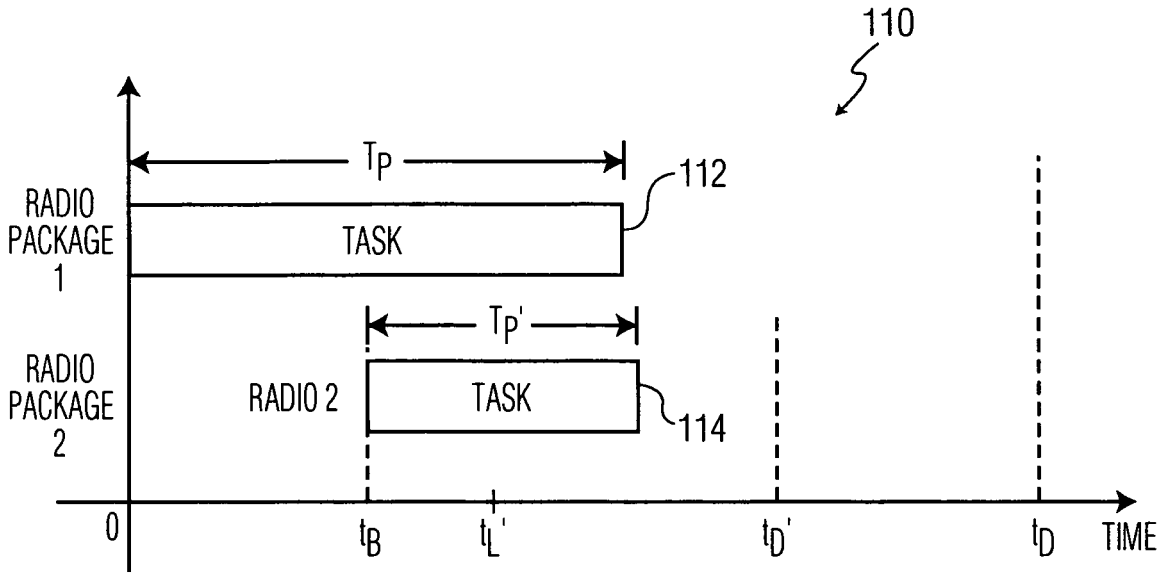

FIG. 3
PRIOR ART

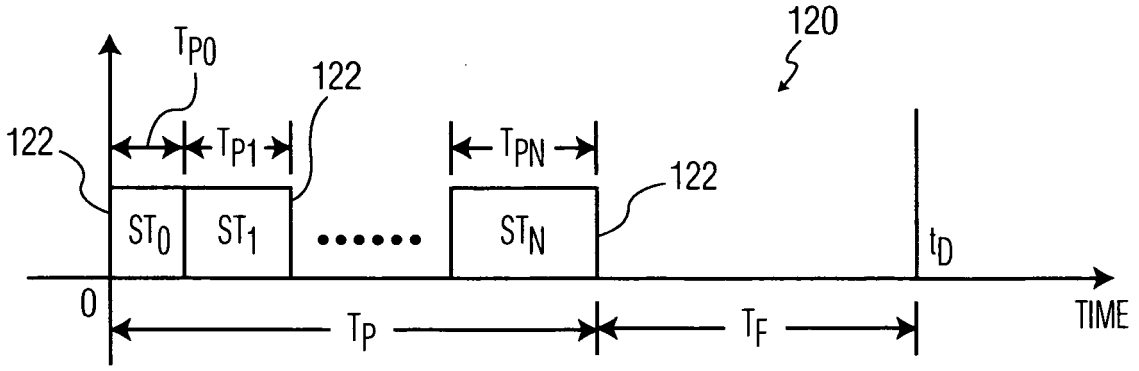

FIG. 4

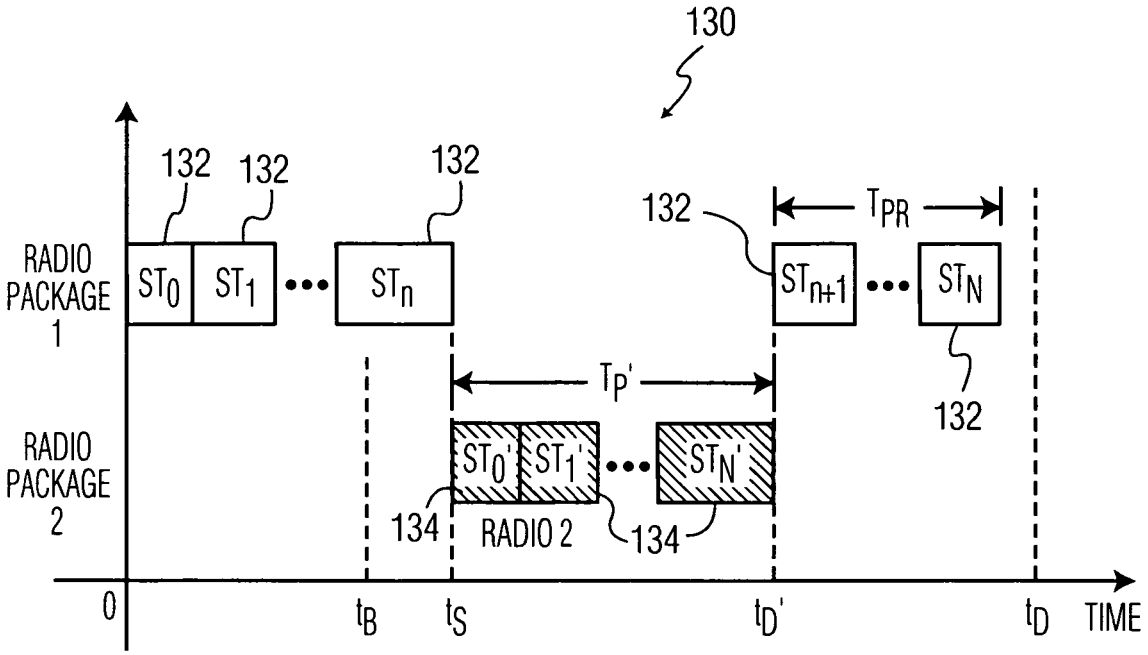
130
132
132
132
132
T_{PR}
RADIO PACKAGE 1
ST_0
ST_1
ST_n
ST_{n+1}
ST_N
T_P'
RADIO PACKAGE 2
ST_0'
ST_1'
ST_N'
134
RADIO 2
134
0
t_B
t_S
t_D'
t_D
TIME

FIG. 5

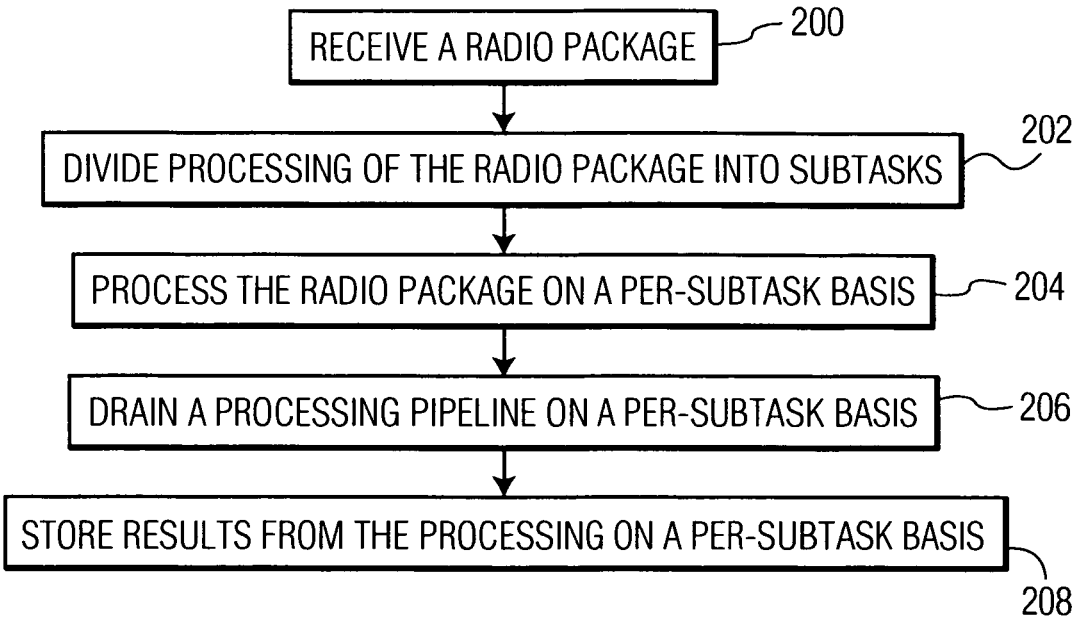
RECEIVE A RADIO PACKAGE
200
DIVIDE PROCESSING OF THE RADIO PACKAGE INTO SUBTASKS
202
PROCESS THE RADIO PACKAGE ON A PER-SUBTASK BASIS
204
DRAIN A PROCESSING PIPELINE ON A PER-SUBTASK BASIS
206
STORE RESULTS FROM THE PROCESSING ON A PER-SUBTASK BASIS
208

FIG. 6

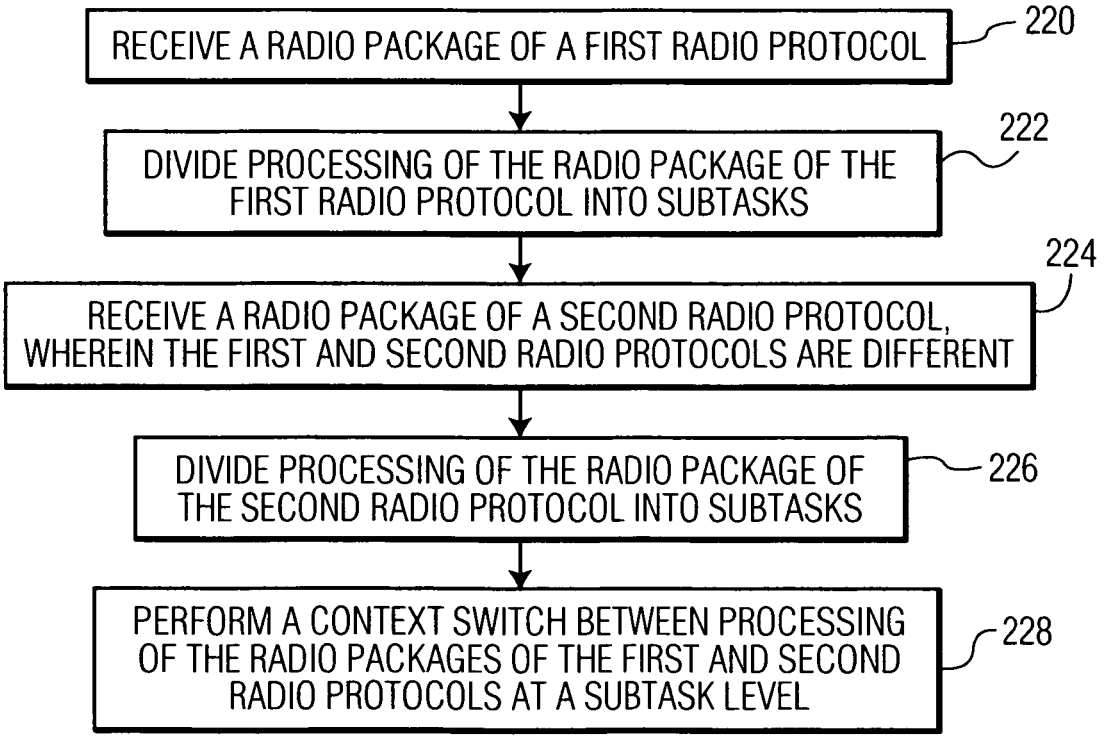
RECEIVE A RADIO PACKAGE OF A FIRST RADIO PROTOCOL
220
DIVIDE PROCESSING OF THE RADIO PACKAGE OF THE FIRST RADIO PROTOCOL INTO SUBTASKS
222
RECEIVE A RADIO PACKAGE OF A SECOND RADIO PROTOCOL, WHEREIN THE FIRST AND SECOND RADIO PROTOCOLS ARE DIFFERENT
224
DIVIDE PROCESSING OF THE RADIO PACKAGE OF THE SECOND RADIO PROTOCOL INTO SUBTASKS
226
PERFORM A CONTEXT SWITCH BETWEEN PROCESSING OF THE RADIO PACKAGES OF THE FIRST AND SECOND RADIO PROTOCOLS AT A SUBTASK LEVEL
228

FIG. 7

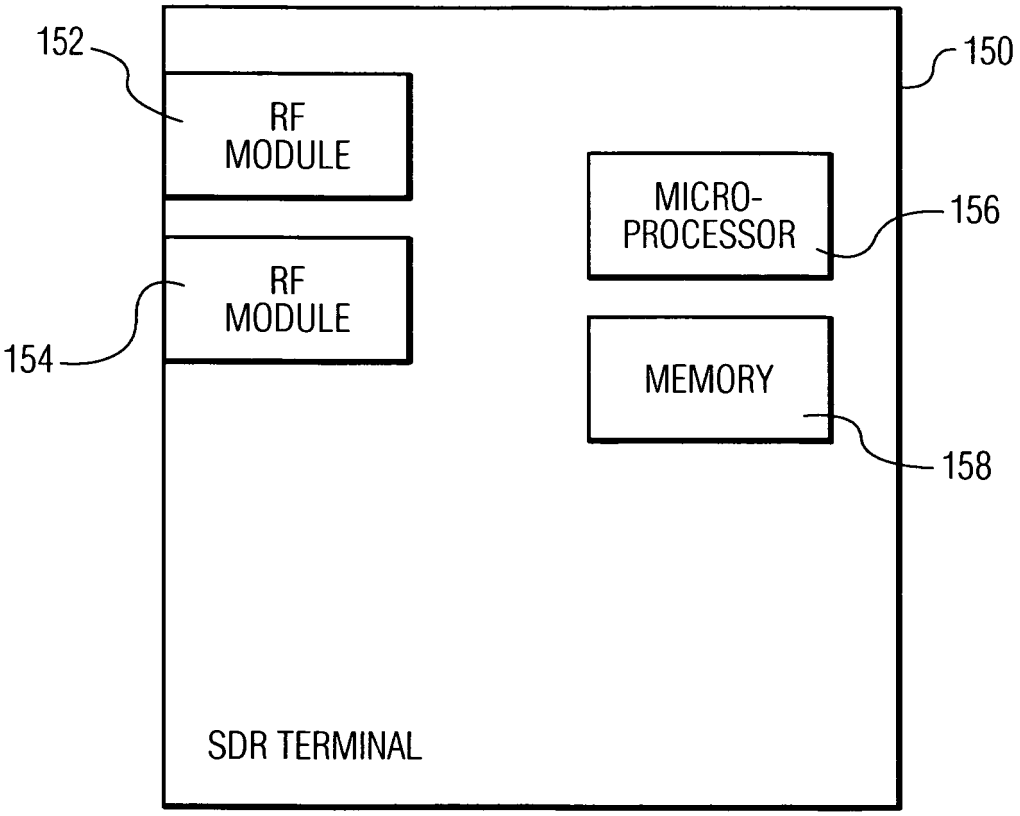
152
RF MODULE
154
RF MODULE
150
MICRO-PROCESSOR
156
MEMORY
158
SDR TERMINAL

FIG. 8

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- WO 2006126943 A **[0010]**